<<EXAM REGISTRATION SYSTEM>>

18CSC209J - Database Management System and Cloud Integration Services

Mini Project Report

Submitted by

VURA VENKATA NAGA SAI BHARGAV ROHITH [Reg. No.: RA2111028010062]
B.Tech. CSE - <<CLOUD COMPUTING>>

A. SUJAN KUMAR REDDY [Reg. No.: RA2111028010032]
B.Tech. CSE - << CLOUD COMPUTING >>

MAHESH REDDY SYAMALA [Reg. No.: RA2111028010056]
B.Tech. CSE - <<CLOUDCOMPUTING>>

DEPARTMENT OF NETWORKING AND COMMUNICATION
SCHOOL OF COMPUTING
COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
(Under Section 3 of UGC Act, 1956)
S.R.M. NAGAR, KATTANKULATHUR – 603 203
KANCHEEPURAM DISTRICT

MAY 2023

BONAFIDE

This is to certify that **18CSC209J - DATABASE MANAGEMENT SYSTEM AND CLOUD INTEGRATION SERVICES LABORATORY Mini Project report** titled **" EXAM REGISTRATION SYSTEM"** is the bonafide work of **VURA VENKATA NAGA SAI BHARGAV ROHITH (RA2111028010062)ALLE SUJAN KUMAR REDDY(RA2111028010032) , MAHESH REDDY SYAMALA(RA2111028010056)** who undertook the task of completing the project within the allotted time.

SIGNATURE
Dr. T.Preethiya
Assistant Professor
Department of Networking and Communication
SRM Institute of Science and Technology

SIGNATURE
Dr.Annapurani.K
Professor and Head
Department of Networking and Communication
SRM Institute of Science and Technology

TABLE OF CONTENTS

ABSTRACT

The aim of Exam Registration System is to automate registering student details, maintaining the student detail, retrieving the specific details, when necessary, security from unauthorized access, functionalities such as insert, delete, update, etc, which must be independent of each other are to be made possible, issuing the hall ticket. This system will help in registering student details to issue hall ticket for the exam. The student can enter to register their details only with their valid username and password. The user can print the hall ticket immediately after registering their details. The administrator will issue the hall ticket only if the student has paid the fee. This Project mainly deals with Login form, Student details, Student report and Student database.

INTRODUCTION

Exam Registration System is an interface between the Student and the Exam Controller responsible for the Issue of Hall Ticket. It aims at improving the efficiency in the Issue of Hall ticket and reduces the complexities involved in it to the maximum possible extent.

PURPOSE

If the entire process of 'Issue of Hall ticket' is done in a manual manner, then it would take several days for the hall ticket to reach the student. Because the number of students for hall ticket is increasing every year, an Automated System becomes essential to meet the demand. So, this system uses several programming and database techniques to elucidate the work involved in this process. As this is a matter of National Security, the system has been carefully verified and validated in order to satisfy it.

SCOPE

• The System provides an online interface to the user where they can fill in their personal details and register for the exam. with the issue of hall ticket can use this system to reduce his workload and process the application in a speedy manner.

• Provide a communication platform between the student and the admin.

• Students will come to know their status of application and the date of exam.

PROBLEM STATEMENT

Exam Registration system.is used in the effective dispatch of registration form to all of the students. This system adopts a comprehensive approach to minimize the manual work and schedule resources, time in a cogent manner. The core of the system is to get the online registration form (with details such as name, reg.no etc.,) filled by the student whose testament is verified for its genuineness by the Exam Registration System with respect to the already existing information in the database. This forms the first and foremost step in the processing of exam application. After the first round of verification done by the system, the information is in turn forwarded to the Exam Controller. The application is then processed manually based on the report given by the system. The system also provides the student the list of exam dates. The controller will be provided with fees details to display the current status of application to the student, which they can view in their online interface. After all the necessary criteria has been met, the original information is added to the database and the hall ticket is sent to the student.

MODULE DESCRIPTION

Module-1: Is the login form in which the user or the admin has to enter the username and password to enter into the exam registration application system.

Module-2: Is the form where the student can enter in to register their details and in the same form the admin can enter to manage the student data base and to retrieve it.

Module-3: Is the form where the student registers their details in order to receive their hall ticket only if they have paid the fee. As soon as the student submits their details the hall ticket will be issued by admin to print if else, the hall ticket will not be issued.

Module-4: Is the form where the admin maintains the student database to insert, delete and update. The complete data of the student can otherwise be viewed in grid form.

USECASE DIAGRAM

The Exam Registration use cases in our system are:

1. Login
2. View exam details
3. Register
4. Acknowledgement
5. Fee Processing

ACTORS INVOLVED:

1. Student
2. System DB

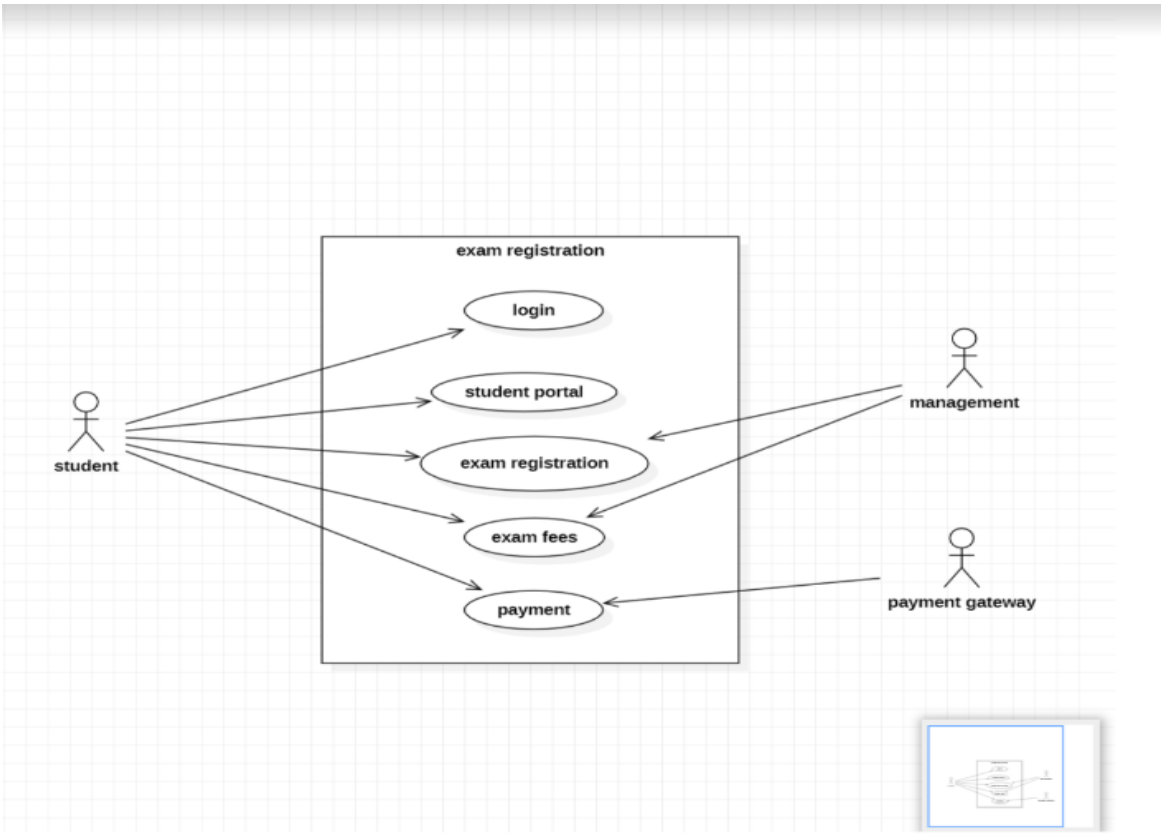
USE-CASE NAME: LOGIN -The student enters his username and password to login and retrieve the information'.

USE-CASE NAME: VIEW EXAM DETAILS - The student views the details about the exam schedule which contains Date, time, etc...

USE-CASE NAME: REGISTER - The student should notify the fee details that only the student can pay the correct amount.

USE-CASE NAME: ACKNOWLEDGEMENT- The exam fees should be paid by the student to get the hall ticket from the exam controller.

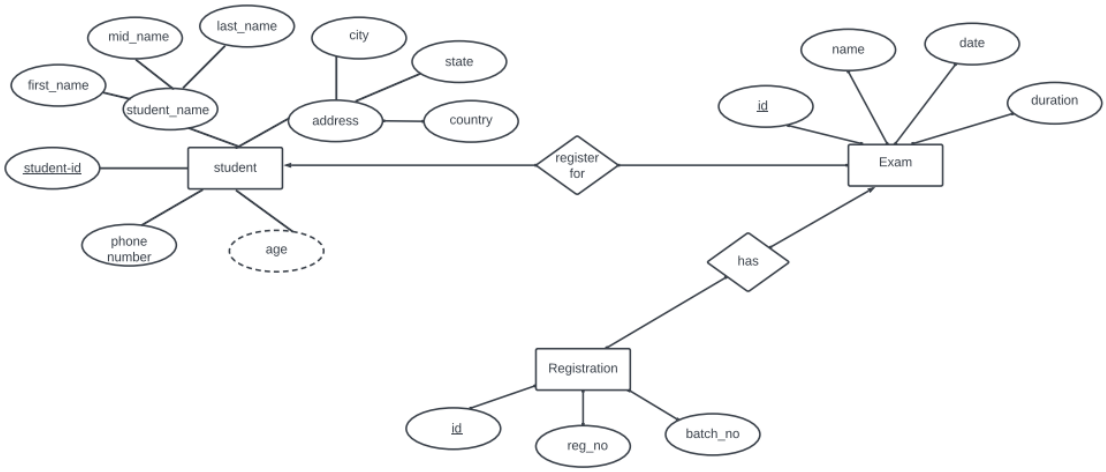
USE-CASE NAME: FEE PROCESSING - All the details should be viewed by both the student and the controller to verify whether all the entered details are correct.



USECASE DIAGRAM FOR EXAM REGISTRATION SYSTEM

ER DIAGRAM

Entity relationship diagram show the relation between different entities such as the relation between student and exam etc.



ER DIAGRAM OF EXAM REGISTRATION SYSTEM

Database Creation using DDL and DML

For Exam Registration System there are three database tables: **Students, Exams and Exam registrations.**

The **students** table stores information about students, the **exams** table stores information about exams, and the **exam registrations** table stores information about which students have registered for which exams and the marks they obtained.

USING DDL

For students

```
CREATE TABLE students (
student_id INT PRIMARY KEY,
student_name VARCHAR(255) NOT NULL,
student_email VARCHAR(255) NOT NULL,
ssn VARCHAR(9) UNIQUE,
student_phone VARCHAR(20) NOT NULL,
student_address VARCHAR(255) NOT NULL
);
```

For Exams

```
CREATE TABLE exams (
exam_id INT PRIMARY KEY,
exam_name VARCHAR(255) NOT NULL,
exam_date DATE NOT NULL,
exam_time TIME NOT NULL,
exam_duration INT NOT NULL
);
```

For Exam Registrations

```
CREATE TABLE exam_registrations (
registration_id INT PRIMARY KEY,
exam_id INT NOT NULL,
student_id INT NOT NULL,
registration_date DATE NOT NULL,
```

registration_time TIME NOT NULL,

FOREIGN KEY (exam_id) REFERENCES exams(exam_id),

FOREIGN KEY (student_id) REFERENCES students(student_id)

);

For courses

CREATE TABLE courses (

course_id INT PRIMARY KEY,

course_name VARCHAR(255) NOT NULL,

instructor_id INT NOT NULL,

instructor_name VARCHAR(255) NOT NULL,

instructor_phone VARCHAR(20) NOT NULL,

duration INT NOT NULL

);

USING DML

For Students

INSERT INTO students (student_id, student_name, student_email,ssn,student_phone, student_address)

VALUES (1, 'Sujan', 'sujan.a@example.com', 'CS032','9177133972', 'Vijayawada'),

(2, 'Bhargav', 'vvnsb.r@example.com', 'CS062', '9676197135', 'Chilakaluripet'),

(3, 'Mahesh Reddy', 'mahesh.s@example.com','CS056', '9765435672', 'Guntur'),

(4, 'Akhil' , 'akhil.j@example.com','CS08', '9985450118', 'Guntur');

For Exams

INSERT INTO exams (exam_id, exam_name, exam_date, exam_time, exam_duration)

VALUES (121, 'Mathematics', '2023-05-15', '09:00:00', 180),

(221, 'Physics', '2023-05-17' , '09:00:00' , 200),

(321, 'Chemistry', '2023-05-20' , '09:00:00' , 120),

(421, 'French' , '2023-05-21' , '10:00:00' ,180);

For Exam Registrations

INSERT INTO exam_registrations (registration_id, exam_id, student_id, registration_date, registration_time)

VALUES (1, 121, 1, '2023-04-20', '12:30:00'),

(2, 221, 2, '2023-04-21', '01:00:00'),

(3, 321, 3, '2023-04-22', '12:00:00'),

(4, 421, 4, '2023-04-22', '12:15:00');

For Course

INSERT INTO courses (course_id, course_name, instructor_id, instructor_name, instructor_phone, duration)

VALUES (101, 'Introduction to calculus and linear algebra', 1234, 'John Smith', '555-1234', 60Days),

(205, 'Introduction to Computer Science', 9112, 'Dr. Alice Brown','522-6162', 45Days),

(303, 'A course in writing and critical thinking', 6534,'Professor Jane Doe', '644-8769', 50Days),

(309, 'A course in Writing and Reading Paragraphs', 7657, 'Professor Ramalingam', '786-9876', 45Days);

Students

student_id	student_name	student_email	ssn	student_phone	student_address
1	Sujan	sujan.a@example.com	CS032	9177133972	Vijayawada
2	Bhargav	vvnsb.r@example.com	CS062	9676197135	Chilakaluripet
3	Mahesh Reddy	mahesh.s@example.com	CS056	9765435672	Guntur
4	Akhil	akhil.j@example.com	CS08	9985450118	Guntur

Exams

exam_id	exam_name	exam_date	exam_time	exam_duration
121	Mathematics	2023-05-15	09:00:00	180
221	Physics	2023-05-17	09:00:00	200
321	Chemistry	2023-05-20	09:00:00	120
421	French	2023-05-21	10:00:00	180

Exam Registration

registration_id	exam_id	student_id	registration_date	registration_time
1	121	1	2023-04-20	12:30:00
2	221	2	2023-04-21	01:00:00
3	321	3	2023-04-22	12:00:00
4	421	4	2023-04-22	12:15:00

For Courses

course_id	course_name	instructor_id	instructor_name	instructor_phone	duration
101	Introduction to calculus and linear algebra	1234	John Smith	555-1234	60Days
205	Introduction to Computer Science	9112	Dr. Alice Brown	522-6162	45Days
303	A course in writing and critical thinking	6534	Professor Jane Doe	644-8769	50Days
309	A course in Writing and Reading Paragraphs	7657	Professor Ramalingam	786-9876	45Days

NORMALIZATION OF DATABASE

A database for an exam registration system would typically have multiple tables to store different types of information, such as students, courses, exam schedules, and registrations. To ensure efficient data management and eliminate data redundancies, normalization of the database is required. The normalization process involves breaking down a table into multiple smaller tables, with each table containing a single type of information. The goal is to minimize data redundancies and inconsistencies and ensure data integrity.

The normalization process typically involves the following steps:

First Normal Form (1NF): Each table should have a primary key, and each column should contain atomic values. This means that each column should hold a single value, rather than a list or set of values.

Second Normal Form (2NF): Each non-key column should be fully dependent on the primary key. This means that there should be no partial dependencies, where a non-key column depends on only a portion of the primary key.

Third Normal Form (3NF): Each non-key column should be dependent only on the primary key, and not on any other non-key columns. This means that there should be no transitive dependencies, where a non-key column depends on another non-key column.

For an exam registration system, the following tables may be required:

Student Table: Contains information about the students, such as name, student ID, and contact details.

Course Table: Contains information about the courses offered, such as course code, course name, and course instructor.

Exam Table: Contains information about the exam schedule, such as exam date, exam time, and exam location.

Registration Table: Contains information about the registration, such as registration ID, student ID and exam date.

Student ID	First Name	SSN	Course	Duration	Instructor	Contact
1	Sujan(S1)	CS032	Maths	60 Days	John Smith	555-1234
2	Bhargav(S2)	CS062	Science	45 Days	Dr. Alice Brown	522-6162
3	Mahesh(S3)	CS056	English	50 Days	Professor Jane Doe	644-8769
4	Akhil(S4)	CS08	French	45 Days	Professor Ramalingam	786-9876

Redundant Data Is Duplicate Data - Course Details And Instructor Details

Student ID	First Name	SSN	Course ID
1	S1	CS032	101
2	S2	CS062	205
3	S3	CS056	303
4	S4	CS08	309

STUDENT TABLE

COURSE TABLE

Course ID	Course	Duration	Instructor ID
101	Maths	60 Days	1234
205	Science	45 Days	9112
303	English	50 Days	6534
309	French	45 Days	7657

Instructor ID	First Name	Contact
1234	John	555-1234
9112	Alice	522-6162
6534	Jane	644-8769
7657	Ramalingam	786-9876

INSTRUCTOR TABLE

Student ID Is Primary Key For Student Table

Course ID Is Primary Key For Course Table And Foreign Key In Student Table

Instructor ID Is Primary Key For Instructor Table And Foreign Key In Course Table

1.Student

- As the attributes of this table does not have sub attributes, it is in first normal form.
- Because every non-primary key attribute is fully functionally dependent on the primary key of the table and it is already in first normal form, this table is now in second normal form.
- Since the table is in second normal form and no nonprimary key attribute is transitively dependent on the primary key, the table is now in 3NF.
- Since the table is in 3NF and no prime key is dependent on non-prime key, so the table is in BCNF.

{student_id} → {first name} (functional dependency exists)

{student_id} → {SSN} (functional dependency exists)

{student_id} → {Course id} (functional dependency exists)

2.Course

- As the attributes of this table does not have sub attributes, it is in first normal form.
- Because every non-primary key attribute is fully functionally dependent on the primary key of the table and it is already in first normal form, this table is now in second normal form.
- Since the table is in second normal form and no nonprimary key attribute is transitively dependent on the primary key, the table is now in 3NF.
- Since the table is in 3NF and no prime key is dependent on non-prime key, so the table is in BCNF.

  {course_id} → {Course} (functional dependency exists)
  {course_id} → {Duration} (functional dependency exists)
  {course_id} → {Instructor_ID} (functional dependency exists)

3.Instructor

- As the attributes of this table does not have sub attributes, it is in first normal form.
- Because every non-primary key attribute is fully functionally dependent on the primary key of the table and it is already in first normal form, this table is now in second normal form.
- Since the table is in second normal form and no nonprimary key attribute is transitively dependent on the primary key, the table is now in 3NF.
- Since the table is in 3NF and no prime key is dependent on non-prime key, so the table is in BCNF.

  {Instructor_id} → {First Name} (functional dependency exists)

{Instructor_id} → {Contact} (functional dependency exists)

IMPLEMENTATION USING DYNAMO DB

To implement an exam registration system using DynamoDB, you can follow the following steps:

1.Define the data model: Determine the data you need to store for the exam registration system. For example, you may need to store information about students, exams, registrations, and results.

2.Create a DynamoDB table: Create a table in DynamoDB to store the data. Define the primary key for the table, which could be a single partition key or a combination of a partition key and a sort key.

3.Define the table schema: Define the attributes that you want to store in the table. You can also specify any secondary indexes that you want to create.

Here Is The Implement using Dynamo DB:

1.Data model:

- **Students**
- ✓ ID
- ✓ Name
- ✓ Email
- **Exams**
- ✓ ID
- ✓ Name
- ✓ Date
- **Registrations**
- ✓ ID
- ✓ Student ID
- ✓ Exam ID
- ✓ Date
- **Results**
- ✓ ID
- ✓ Student ID
- ✓ Exam ID
- ✓ Score

2.DynamoDB table:

- Table name: ExamRegistration
- Primary key: Partition key - ID (string)

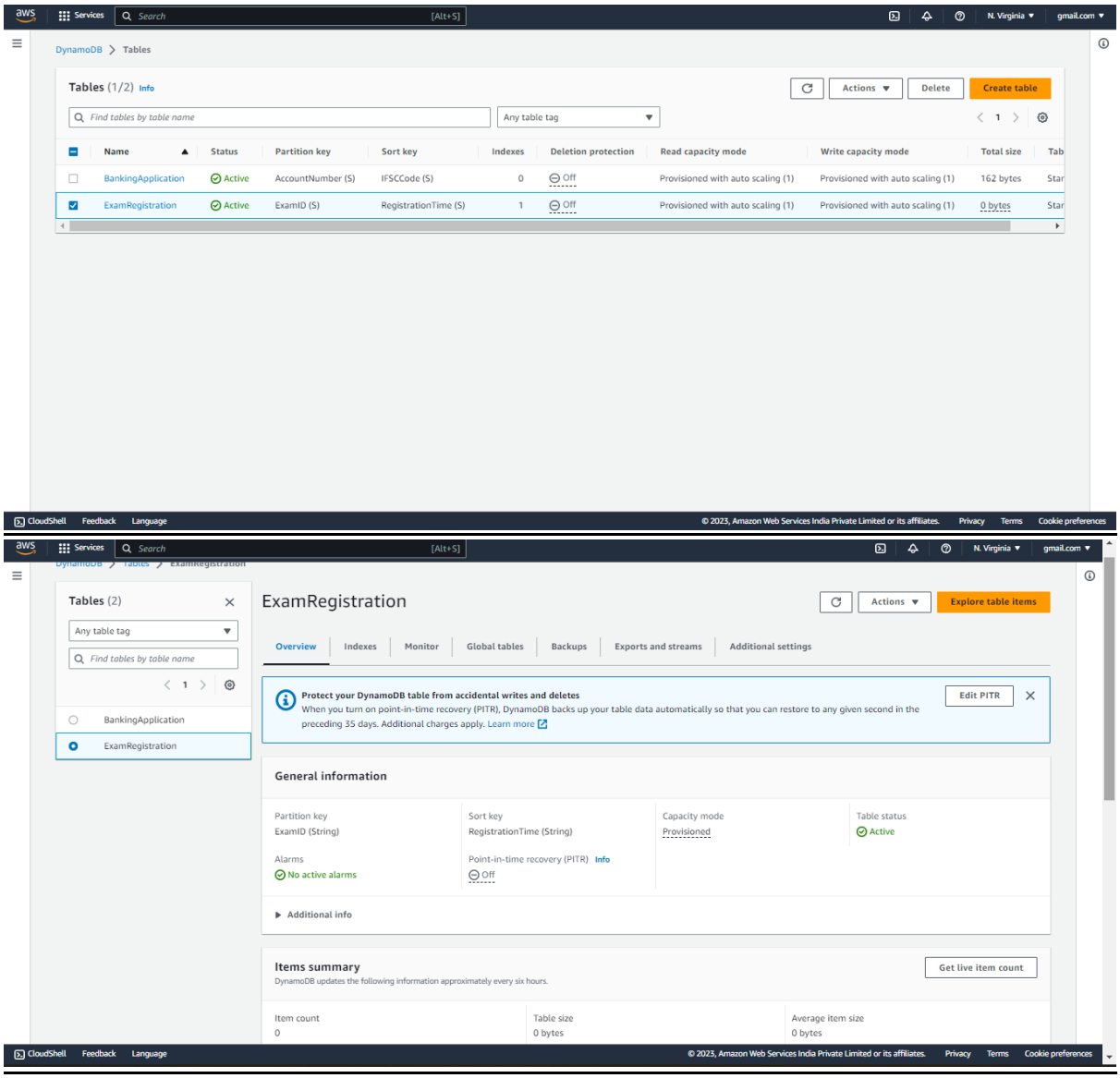
3.Table schema:

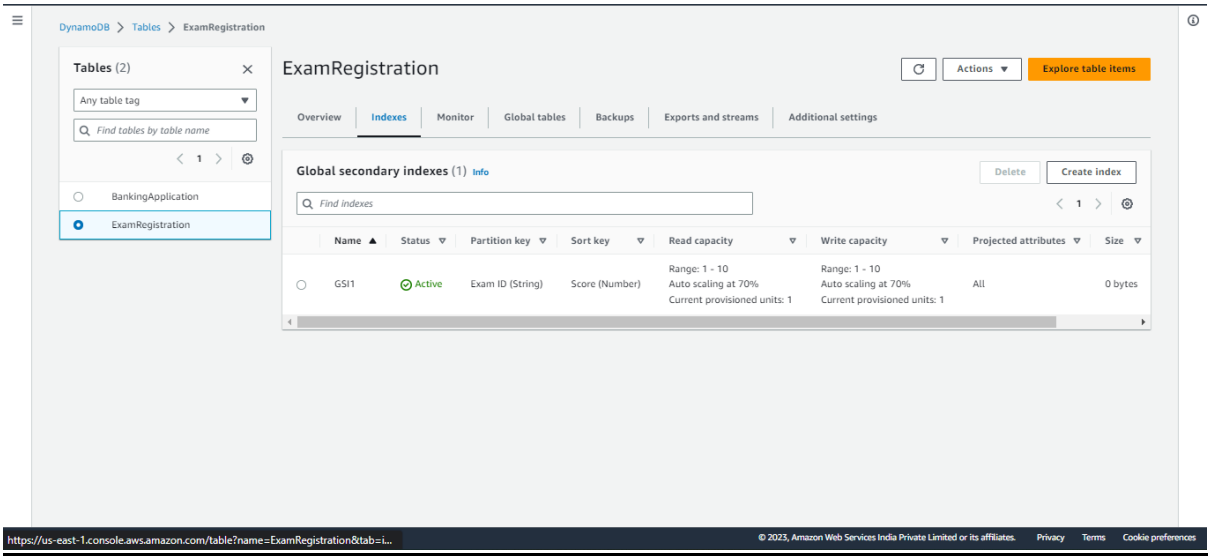
- **Attributes:**

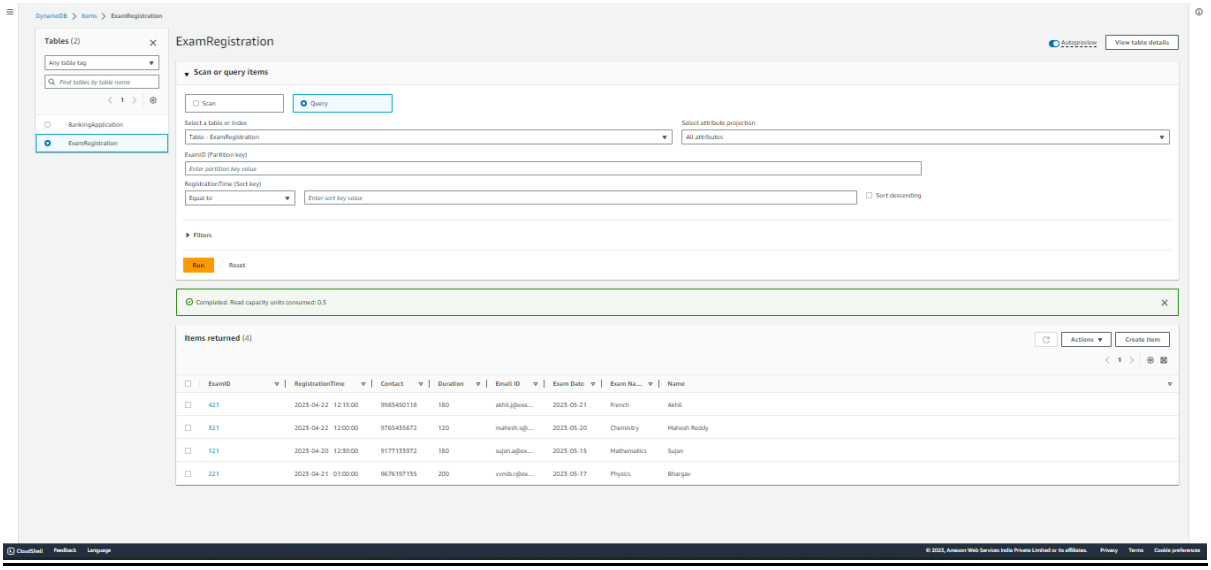
✓ ID (string)
✓ Name (string)
✓ Email (string)
✓ Exam ID (string)
✓ Exam Name (string)
✓ Exam Date (string)
✓ Registration Date (string)
✓ Score (number)

Secondary indexes:

- GSI1: Partition key - Exam ID (string), Sort key - Score (number)







{
"ExamID": {
"S": "121"
},
"RegistrationTime": {
"S": "2023-04-20\t12:30:00"
},
"Contact": {
"S": "9177133972"
},
"Duration": {
"N": "180"
},
"Email ID": {
"S": "sujan.a@example.com"
},

```
"Exam Date": {
"S": "2023-05-15"
},
"Exam Name": {
"S": "Mathematics"
},
"Name": {
"S": "Sujan"
}
}
{
"ExamID": {
"S": "421"
},
"RegistrationTime": {
"S": "2023-04-22\t12:15:00"
},
"Contact": {
"N": "9985450118"
},
"Duration": {
"N": "180"
},
"Email ID": {
"S": "akhil.j@example.com"
},
"Exam Date": {
"S": "2023-05-21"
},
"Exam Name": {
"S": "French"
},
"Name": {
"S": "Akhil"
}
}
{
"ExamID": {
"S": "221"
},
"RegistrationTime": {
"S": "2023-04-21\t01:00:00"
},
"Contact": {
"N": "9676197135"
```

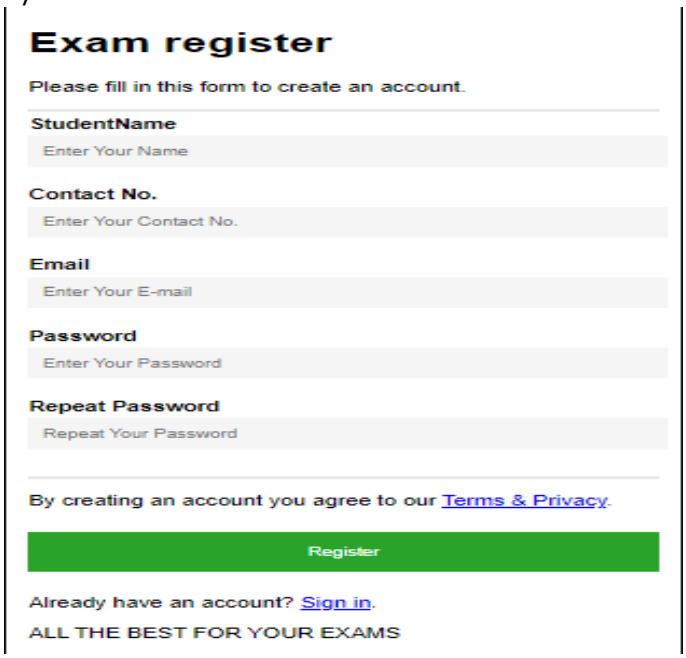
```
},
"Duration": {
"N": "200"
},
"Email ID": {
"S": "vvnsb.r@example.com"
},
"Exam Date": {
"S": "2023-05-17"
},
"Exam Name": {
"S": "Physics"
},
"Name": {
"S": "Bhargav"
}
}
{
"ExamID": {
"S": "321"
},
"RegistrationTime": {
"S": "2023-04-22\t12:00:00"
},
"Contact": {
"S": "9765435672"
},
"Duration": {
"N": "120"
},
"Email ID": {
"S": "mahesh.s@example.com"
},
"Exam Date": {
"S": "2023-05-20"
},
"Exam Name": {
"S": "Chemistry"
},
"Name": {
"S": "Mahesh Reddy"
}
}
```

CODE:

```
<!DOCTYPE html>
<html>
<head>
<style>
body
{
font-family: Arial, Helvetica, sans-serif;
margin-left:25%;
margin-right:25%;
border: 1px solid #000000;
margin-bottom: 5px;
padding: 0px 15px 0 15px;
}
input[type=text], input[type=password]
{
width: 97%;
padding: 10px;
margin: 5px 0 22px 0;
display: inline-block;
border: none;
background: #F5F5F5;
}
hr
{
border: 1px solid #e6e6e6;
margin-bottom: 5px;
}
.registerbutton
{
background-color: #29a329;
color: white;
padding: 15px 20px;
margin: 10px 0px;
border: none;
cursor: pointer;
width: 100%;
text:bold;
}


</style>
</head>
<body>
<form action="action.php">
<h1>Exam register</h1>
```

```
<p>Please fill in this form to create an account.</p>
<hr>
<label for="name"><b>StudentName</b></label>
<input type="text" placeholder="Enter Your Name" name="name" required><br>
<label for="number"><b>Contact No.</b></label>
<input type="text" placeholder="Enter Your Contact No." name="number"
required><br>
<label for="email"><b>Email</b></label>
<input type="text" placeholder="Enter Your E-mail" name="email" required><br>
<label for="pwd"><b>Password</b></label>
<input type="password" placeholder="Enter Your Password" name="psw"
required><br>
<label for="psw-repeat"><b>Repeat Password</b></label>
<input type="password" placeholder="Repeat Your Password" name="psw-repeat"
required><br>
<hr>
<p>By creating an account you agree to our <a href="#">Terms & Privacy</a>.</p>
<button type="submit" class="registerbutton">Register</button>
<p>Already have an account? <a href="#">Sign in</a>.</p>
<p>  ALL THE BEST FOR YOUR EXAMS  <a
</form>
</body>
</html>
```

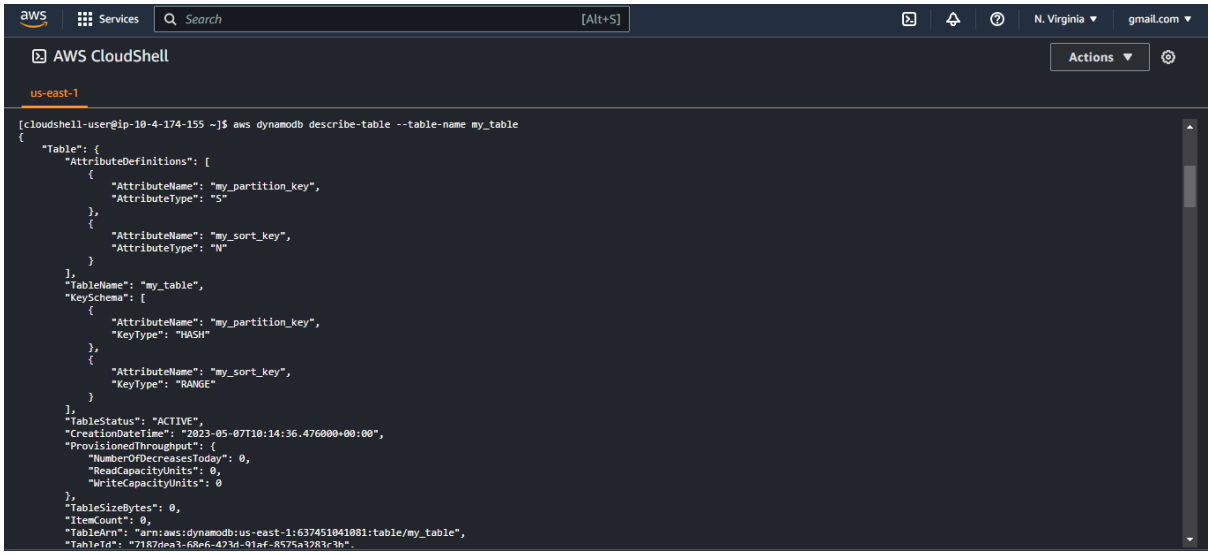


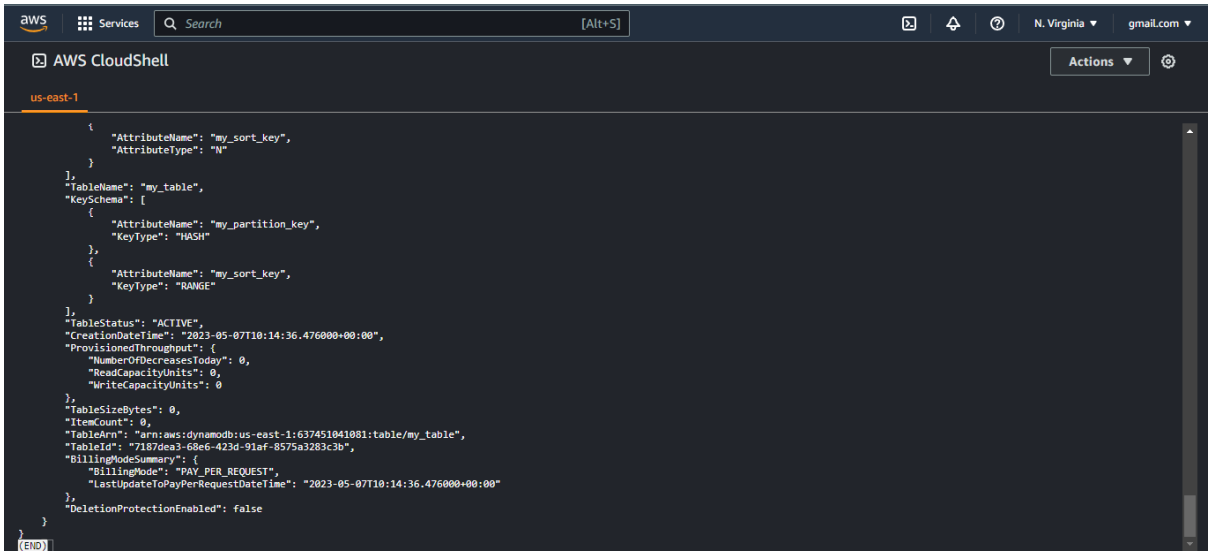
CONCLUSION

In conclusion, the exam registration system project is an essential tool for managing student registration and exam schedules. It helps to simplify the registration process by providing a user-friendly interface and ensures that students can register for their exams without any complications. The system can handle large

amounts of data and store them securely. The system can also generate reports and analyse data to provide useful insights to the management team.
Overall, the exam registration system can improve the efficiency of the registration process and enhance the overall student experience.

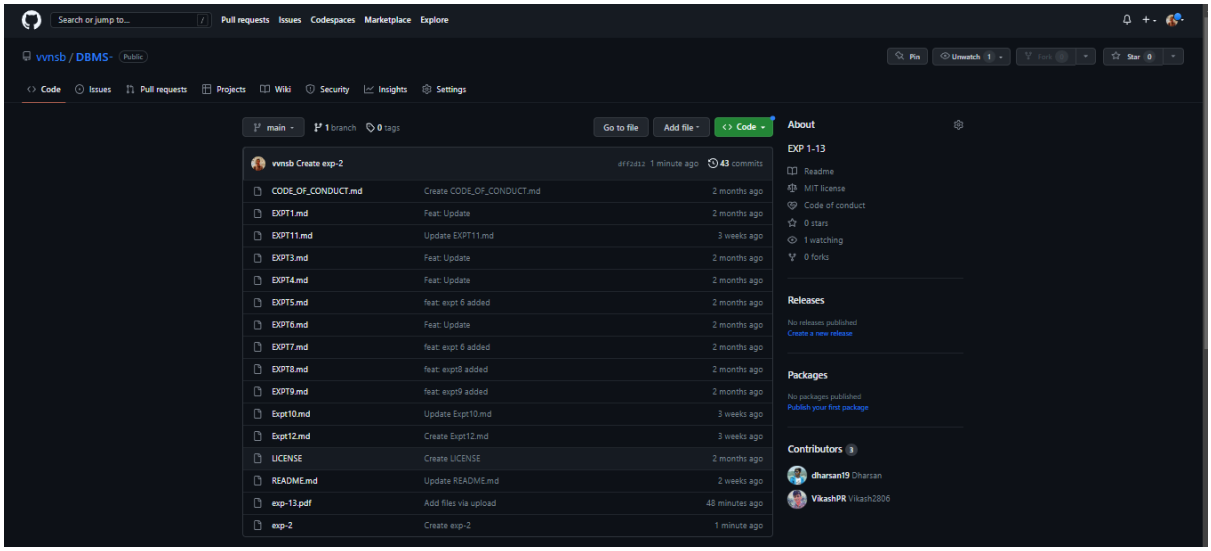
Appendix I – Screenshot



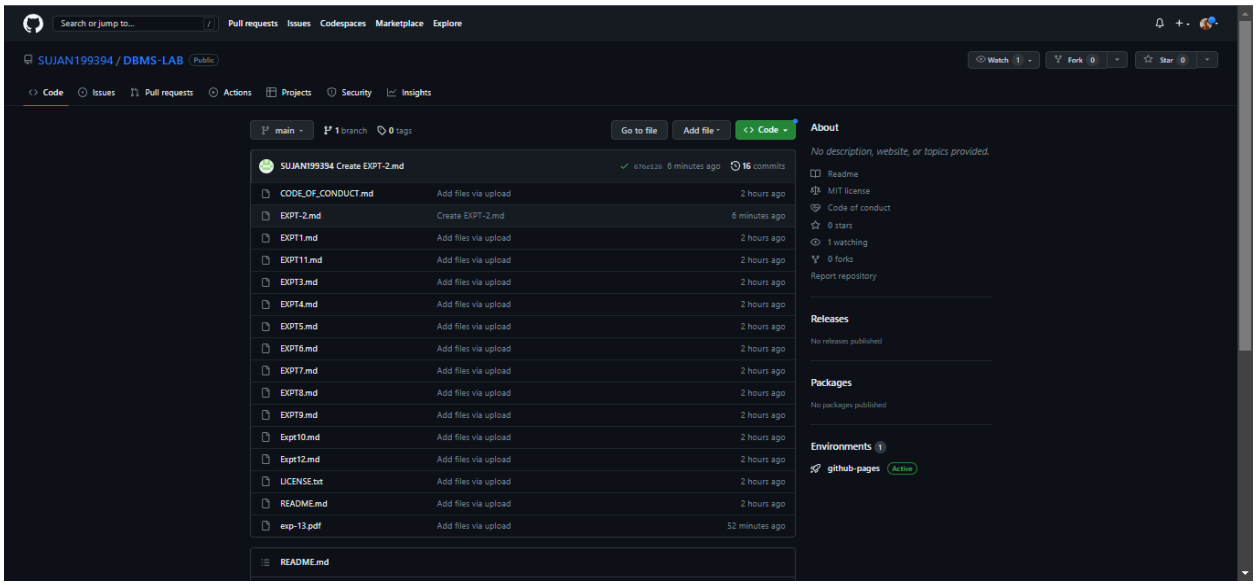


Appendix II - GitHub Profile

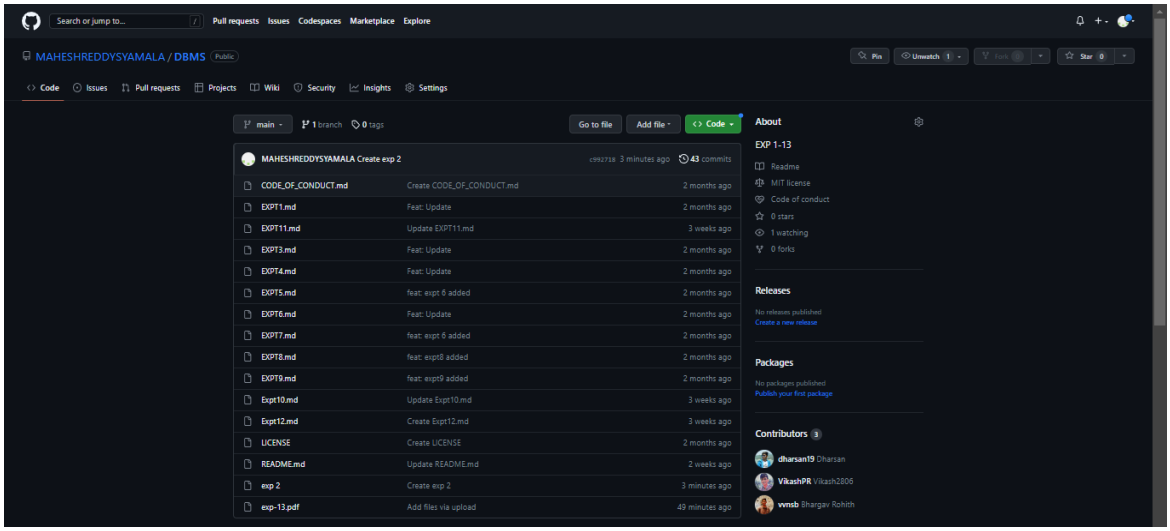
https://github.com/vvnsb



https://github.com/SUJAN199394



https://github.com/MAHESHREDDYSYAMALA



Appendix III - AWS Course Completion certificate


aws academy
SUJAN KUMAR REDDY ALLE
(RA2111028010032)
Certificate of Completion for
AWS Academy Graduate - AWS Academy Cloud Architecting
Course hours completed
40 hours
Issued on
04/22/2023
Digital badge
https://www.credly.com/go/GofLVMRQ


aws academy
MAHESH REDDY SYAMALA
Certificate of Completion for
AWS Academy Graduate - AWS Academy Cloud Architecting
Course hours completed
40 hours
Issued on
04/23/2023
Digital badge
https://www.credly.com/go/SUfj0fef

DATABASE MANAGEMENT SYSTEMS AND CLOUD INTEGRATION SERVICES – 18CSC109J

B.Tech CSE (Cloud Computing)

Student Portfolio

Name : Vura Venkata Naga Sai Bhargav Rohith

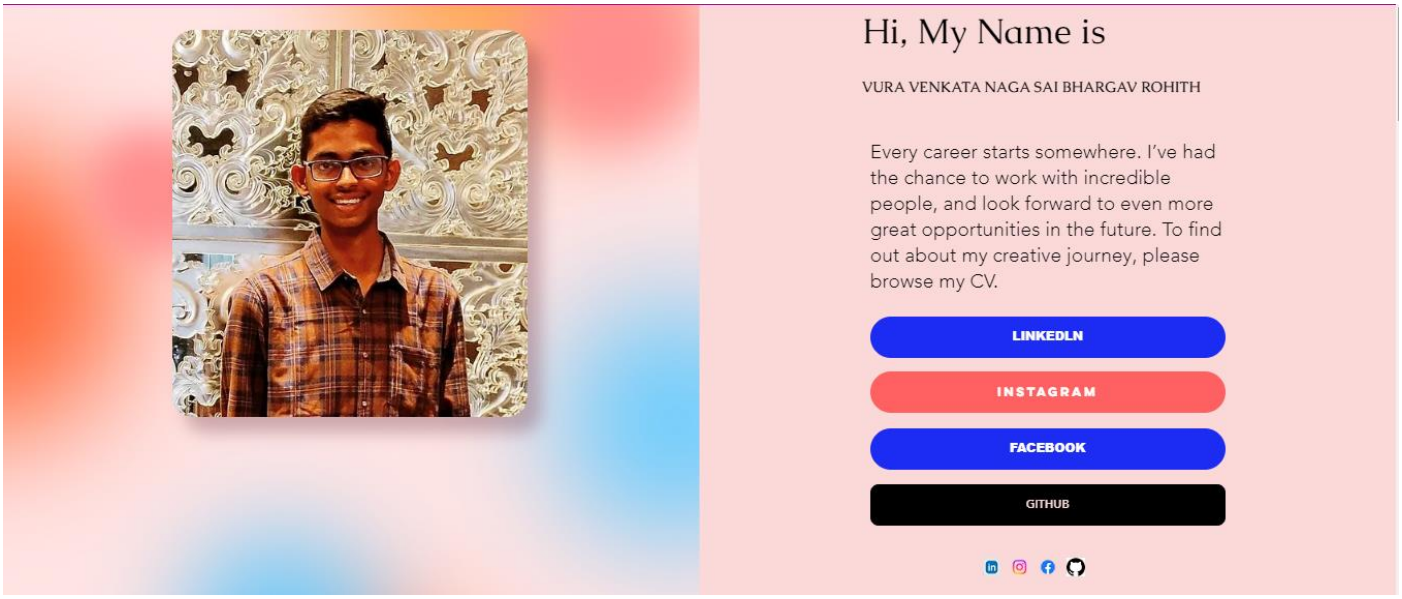
Reg No : RA2111028010062

Year : 2nd year

Section : O1

Course Teacher : Dr. T.Preethiya

Portfolio link: https://vvnsbrohith2019.wixsite my-site-3.com/



List of Assignments:

- Assignment 1 – DDL Queries
- Assignment 2 – SQL/PLSQL Procedure

VVNSB Rohith

Signature of the student

DATABASE MANAGEMENT SYSTEMS AND CLOUD INTEGRATION SERVICES – 18CSC109J

B.Tech CSE (Cloud Computing)

Student Portfolio

Name : Alle Sujan Kumar Reddy

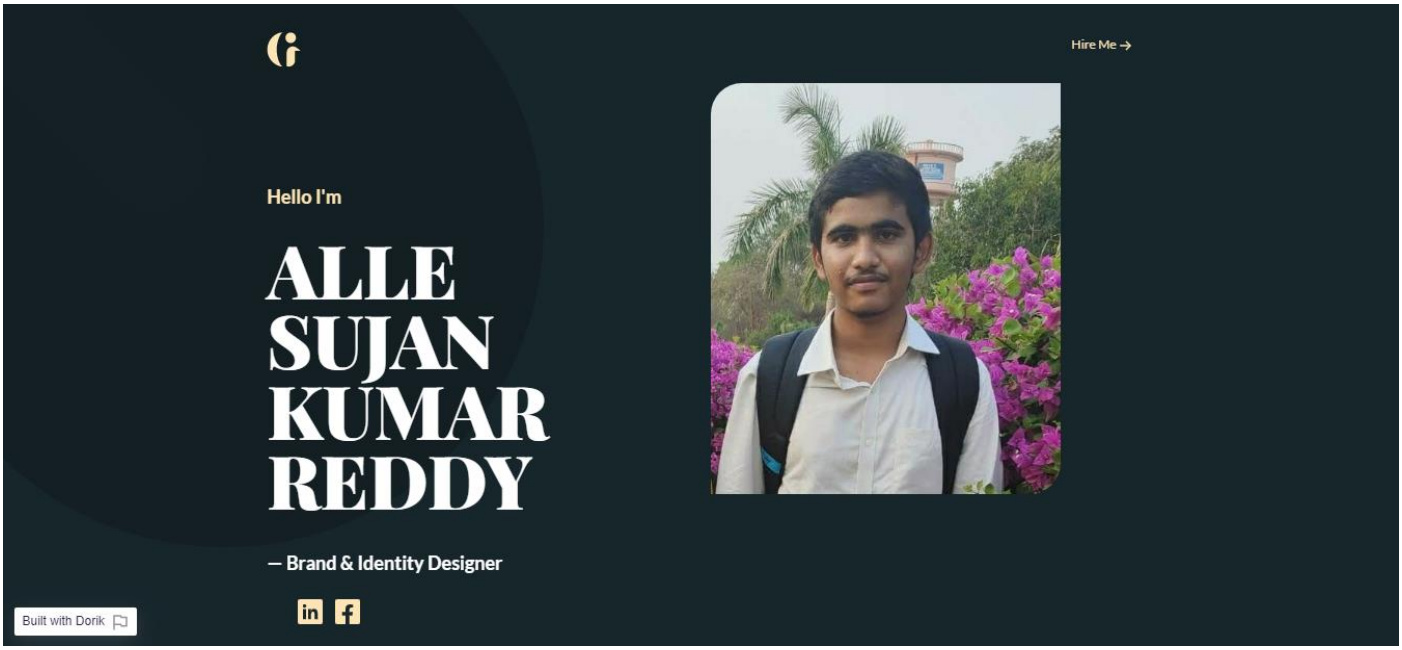
Reg No : RA2111028010032

Year : 2nd year

Section : O1

Course Teacher : Dr. T.Preethiya

Portfolio link: https://sujan.dorik.io/



List of Assignments:

- Assignment 1 – DDL Queries
- Assignment 2 – SQL/PLSQL Procedure

Alle Sujan Kumar Reddy

Signature of the student

DATABASE MANAGEMENT SYSTEMS AND CLOUD INTEGRATION SERVICES – 18CSC109J

B.Tech CSE (Cloud Computing)

Student Portfolio

Name : Mahesh Reddy Syamala

Reg No : RA2111028010056

Year : 2nd year

Section : O1

Course Teacher : Dr. T.Preethiya

Portfolio link: https://mahesh.dorik.io/

List of Assignments:

- Assignment 1 – DDL Queries
- Assignment 2 – SQL/PLSQL Procedure

Mahesh Reddy Syamala

Signature of the student

V V N S B Rohith
RA2111028010062
01

1. Write the DDL query to Create a table name of Student 1
S.No, Name, Stipend, Stream, Average Marks, Grade, Class.

2. Insert the data into table.

3. Updat the stream of S.NO (5).

4. Delete row no 7 use drop and trancate where ever neccessary.

5. Commit your changes. Create save point and ROLL BACK.

---

1. CREATE TABLE STUDENT1
(
S.NO, Number;
Name, String;
Stipend, Int;
Stream Varchar(30);
Average Marks Int;
Grade Varchar(30);
Class Varchar(30);
);

S.NO	Name	Stipend	Stream	Average Marks	Grade	Class
1.	Rohith	40,000	Medicine	80	A	01
2.	Akhil	20,000	CSE	70	A	1st year
3.	Rahul	10,000	Engineering	86	A	1st year
4.	Anil	20,000	Mechanical	76	A	2nd year
5.	Mahesh	10,000	CSE	96	A	2nd year
6.	Sujan	20,000	Medicine	70	A	01
7.	Krishna	30,000	CSE	40	A	1st year
8.	Bhasha	20,000	Agriculture	50	A	2nd year
9.	Suresh	10,000	CSE	80	A	2nd year
10.	Atthram	40,000	Medicine	70	A	01

2. Insert into Student1(1, 'Rohith', '40,000', 'Medicine', 80, A, 01);
Insert into Student1(2, 'Akhil', '20,000', 'CSE', 70, A, 1st year);
Insert into Student1(3, 'Rahul', '10,000', 'Engineering', 86, A, 1st year);
Insert into Student1(4, 'Anil', '20,000', 'Mechanical', 76, A, 2nd year);
Insert into Student1(5, 'Mahesh', '10,000', 'CSE', 96, A, 2nd year);
Insert into Student1(6, 'Sujan', '20,000', 'Medicine', 70, A, 01);
Insert into Student1(7, 'Krishna', '30,000', 'CSE', 40, A, 1st year);
Insert into Student1(8, 'Bhasa', '20,000', 'Agriculture', 50, A, 2nd year);
Insert into Student1(9, 'Suresh', '10,000', 'CSE', '80', A, 2nd year);
Insert into Student1(10, 'Abhiram', '40,000', 'Medicine', 70, A, 01);

3. Update Student 1 Set Stream = 'Law' where S.NO = 5;

4. Delete from Student 1 where S.NO = 7;

5. COMMIT TRANSACTION;
DROP TABLE STUDENT1;

13/23

Name: A. Sujan Kumar Reddy
Regno: RA2111028010032

) Write the DDL query to create a table the name of student

S.No	name	Stipend	stream	Average marks	grade	class
1.	Rohit	1000	CSE	90	A	01
2.	mahesh	2000	CSE	91	A	01
3.	Sathvik	3000	mechanical	90	B	01
4.	Sharan	4000	medical	94	A	01
5.	Rishi	2000	CSE	92	B	01
6.	Raju	4000	medical	96	O	01
7.	Bhanu	1000	Aerospace	92	A	01
8.	Krishna	2000	mechanical	94	B	01
9.	Eshwar	3000	CSE	93	B	01
10.	Karthik	2000	CSE	90	A	01

2. ~~write~~ update Stream of Sl.no 5 to Some other values
3. Delete roll no :- 7 use drop and truncate where ever necessary
4. Commit your change, create Save point and roll back

---

1. create table student1(
   Sl.no int,
   name varchar(10),
   Stipend int,
   Stream varchar(10),
   avg-mark int,
   grade varchar(2),
   class varchar(1));

insert into student (1, "Rohit", "1000", "CSE", 90, "A", "01");
insert into student (2, "mahesh", 2000, "CSE", 91, "A", "01");
insert into student (3, "Sathvik", 3000, "mechanical", 90, "B", "01");
insert into student (4, "Sharan", 4000, "medical", 94, "A", "01");
insert into student (5, "Rishi", 2000, "CSE", 92, "B", "01");
insert into student (6, "Raju", 4000, "medical", 96, "O", "01");

insert into student (7, "Bhanu", 1000, "Aerospace", 95, "B", "01");
insert into student (8, "krishna", 2000, "mechinical", 92, "B", "01");
insert into student (9, "Eshwar", 3000, "CSE", 93, "B", "01");
insert into student (10, "karthik", 2000, "CSE", 90, "A", "01");

2) update student
set stream = "medical"
where stream = "CSE";

3) Delete from student
where sl.no = 7;

4) commit Transaction;
drop table student;

13/23

MAHESH REDDY
RA2111028010056

DDL

1. write Query to create table name of student

S.No Name stephen stream avg.mark grade class

Insert to datatype

Update the stream of values of 5

Delete Row 7 use drop and drunket where necessary.

Commit your changes. create same points. Roll back

---

1. CREATE TABLE STUDENT 1

```
(
    S.NO  Number(20),
    Name  Varchar(60),
    Stephen  Int(20),
    Stream  Varchar(30),
    avg.mark  Int,
    Grade  Varchar(20),
    Class  Varchar(30);
);
```

S.NO	Name	stephen	stream	avg.mark	grade	Class
1	Trishan	60,000	Medical	86	O	P1
2	shashank	80,000	CSE	90	O+	O1
3	siddranth	80,000	MBA	84	O	O2
4	Kamal	90,000	CSE	95	O+	A1, B1
5	Budata	75,000	CSE	80	O	O1
6	Rohith	86,000	LAW	75	A+	LC1
7	Mahesh	39,000	CSE	84	O	O1
8	sanjay	95,000	BBA	70	A+	2nd year
9	SUJAN	70,00	CSE	84	A+	O1
10	Babbrao	6,000	BIO	50	C	7th year

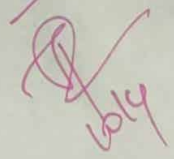
(i) Insert Into student (1, 'Trishan', 60,000, Medical, 86, O, P1);
Insert Into student (2, 'shashank', 80,000, CSE, 90, O+, O1)
Insert Into student (3, 'siddranth', 80,000, MBA, 84, O, O2)
Insert Into student (4, 'Kamal', 90000, CSE, 95, O, A1, B1)
Insert Into student (5, 'Budata', 75,000, CSE, 80, O, O1)
Insert Into student (6, 'Rohith', 86,00, LAW, 75, A+, LC1)
Insert Into student (7, 'Mahesh', 39,000, CSE, 84, O, O1)
Insert Into student (8, 'sanjay', 95,000, BBA, 70, A+, 2nd year)
Insert Into student (9, 'SUJAN', 70,000, CSE, 84, A+, O1)
Insert Into student (10, 'Subbarao', 6000, BIO, 50, C, 7th year)

(ii) Update student 1 set stream = 'BIO' where S.NO = 5;

VVNSB Rohith
RA2111028010062
01

→ Product of two numbers using functions.

```
DECLARE
a number;
b number;
answer number;
Function product (a number, b number)
Return Number
AS C Number;
Begin
C:=A*B;
Return (C);
End Product;
Begin
  A:=&A;
  B:=&B;
Answer:=product (A,B);
DBMS_Output.Put_line ("The Product of two Numbers
                       is "||Answer);
End;
/
```



MAHESH REDDY
RA2111028010056

→ Calculating Factorial : (Reursive)

```
Declare
   num number;
factorial number;
Function fact (x number)
Return number
Is
     f number;
Begin
     if x=0 then
          fi=1;
     else
      fi=x * fact (x-1);
     End if;
     Return f;
     End;
}
```

Name : A. Sujan Kumar Reddy
Reg no : RA2111028010032

```
Declare
n number:
i number:
temp number;
Begin
n := 13;
i := 2;
temp := 1;
for i in 2...n/2
loop
    if mod (n,i) = 0
    then
      temp := 0;
      exit;
    end if;
end loop;
if temp = 1
then
    dbms_output.put_line ('true');
else
    dbms_output.put_line ('false');
end if;
end;
-- Program End
```

Input : 5
Output : true